

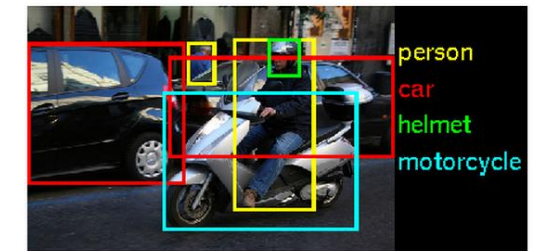
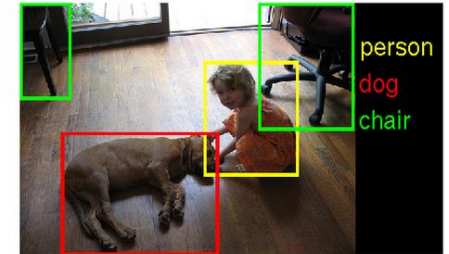
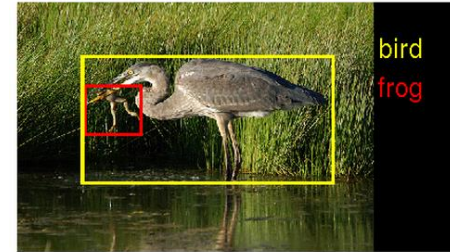
# Toward Accelerating Deep Learning at Scale Using Specialized Hardware in the Datacenter

Kalin Ovtcharov, Olatunji Ruwase, Joo-Young Kim,  
Jeremy Fowers, Karin Strauss, Eric S. Chung

Acknowledgments: Doug Burger and the Catapult Team, Trishul Chilimbi and the Digital Cortex Team, Altera Corporation

# The Rise of Deep Learning

- Significant advances in
  - Computer vision
  - Speech recognition
  - Natural language processing
  - Recommendation systems
  - Intelligent agents
  - Etc.
- Examples
  - Convolutional Neural Networks (CNNs)
  - Deep Belief Networks (DBNs)
  - Recurrent Neural Networks (RNNs)
  - ... ?



## Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification

Kaiming He    Xiangyu Zhang    Shaoqing Ren    Jian Sun

Microsoft Research  
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

### Abstract

*Rectified activation units (rectifiers) are essential for state-of-the-art neural networks. In this work, we study rectifier neural networks for image classification from two aspects. First, we propose a Parametric Rectified Linear Unit (PReLU) that generalizes the traditional rectified unit. PReLU improves model fitting with nearly zero extra computational cost and little overfitting risk. Second, we de-*

*and the use of smaller strides [33, 24, 2, 25]), new non-linear activations [21, 20, 34, 19, 27, 9], and sophisticated layer designs [29, 11]. On the other hand, better generalization is achieved by effective regularization techniques [12, 26, 9, 31], aggressive data augmentation [16, 13, 25, 29], and large-scale data [4, 22].*

*Among these advances, the rectifier neuron [21, 8, 20, 34], e.g., Rectified Linear Unit (ReLU), is one of several keys to the recent success of deep networks [16]. It expe-*

# This Talk:

## Are FPGAs a Promising Target in the Datacenter for Deep Learning<sup>1</sup>?

<sup>1</sup>*Case study: CNN-based Image Classification (inference)*

# Cloud Specialization Tradeoffs



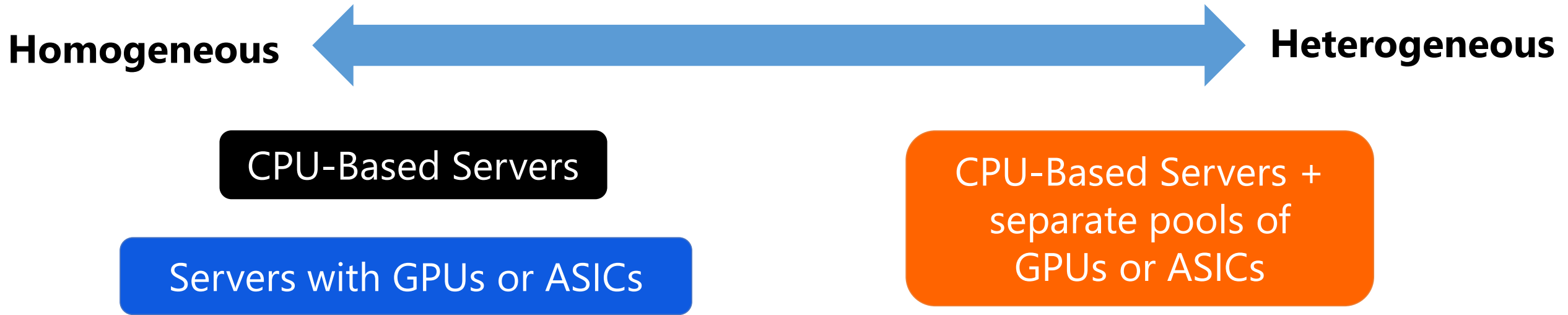
- + Excellent maintainability in datacenter
- + Maximum flexibility for all workloads
- Performance of CNNs/DNNs vastly slower than specialized HW

# Cloud Specialization Tradeoffs



- + CNNs/DNNs that utilize GPUs or ASICs benefit significantly
- CNNs/DNNs cannot scale beyond limited pools
- Heterogeneity challenging for maintainability

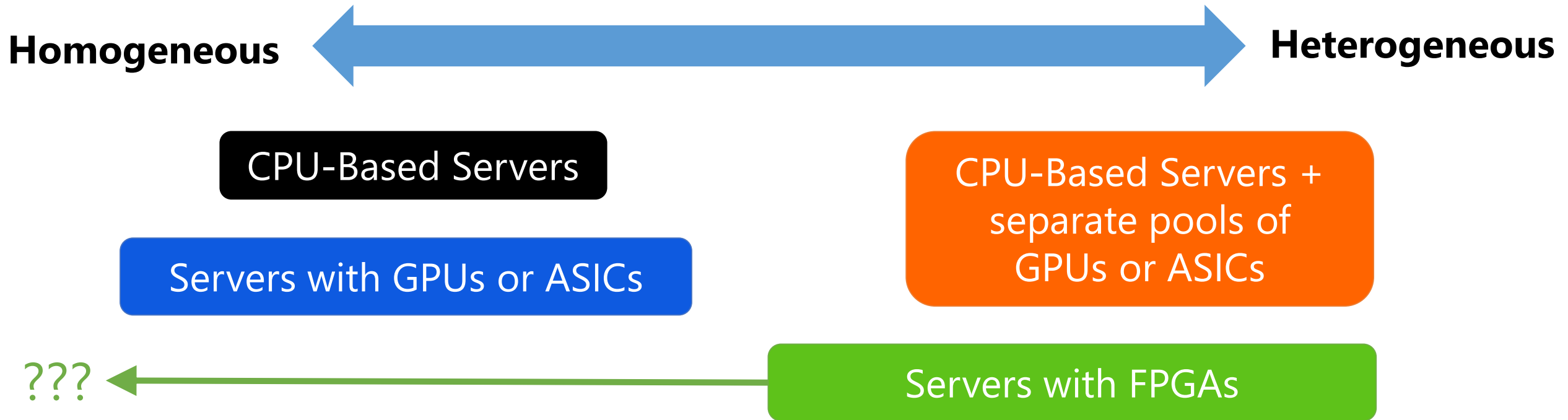
# Cloud Specialization Tradeoffs



+ Homogeneous

- Increased cost and power per server (particularly GPUs)
- Not economical for all applications in the datacenter (GPUs and ASICs)

# Cloud Specialization Tradeoffs

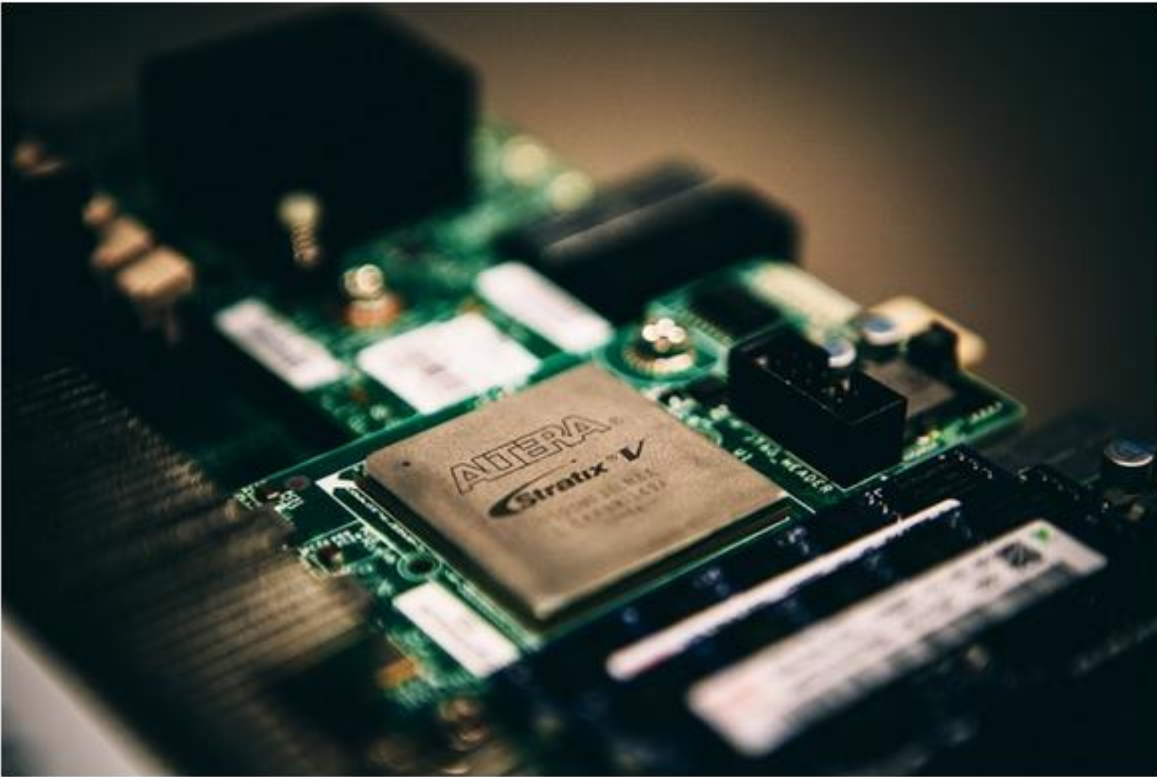


- + Homogeneous
- + Low overhead in power and cost per server
- + Flexibility benefits many workloads?

- Lower peak performance than GPUs or ASICs on some workloads

➔ ***Overtake through scale?***

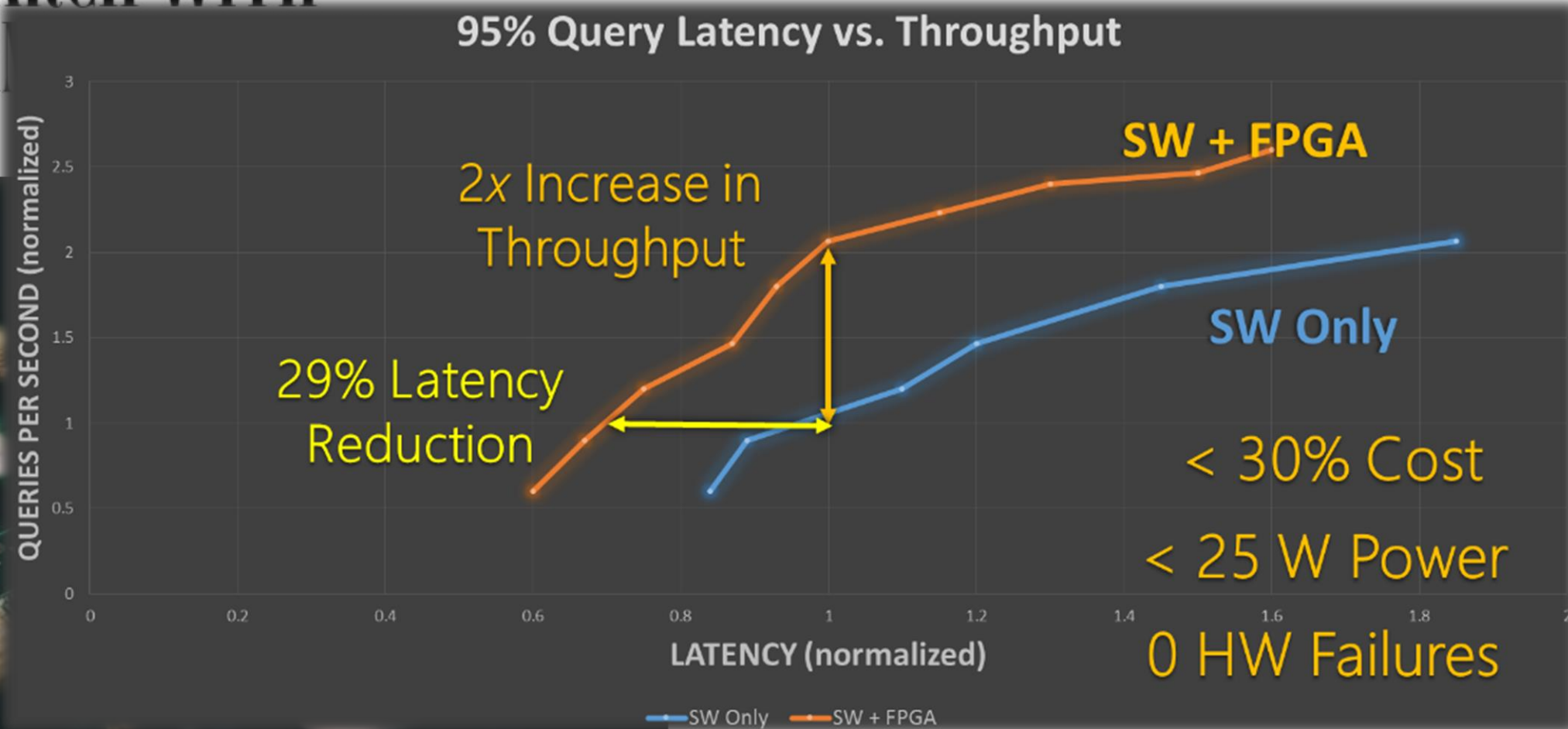
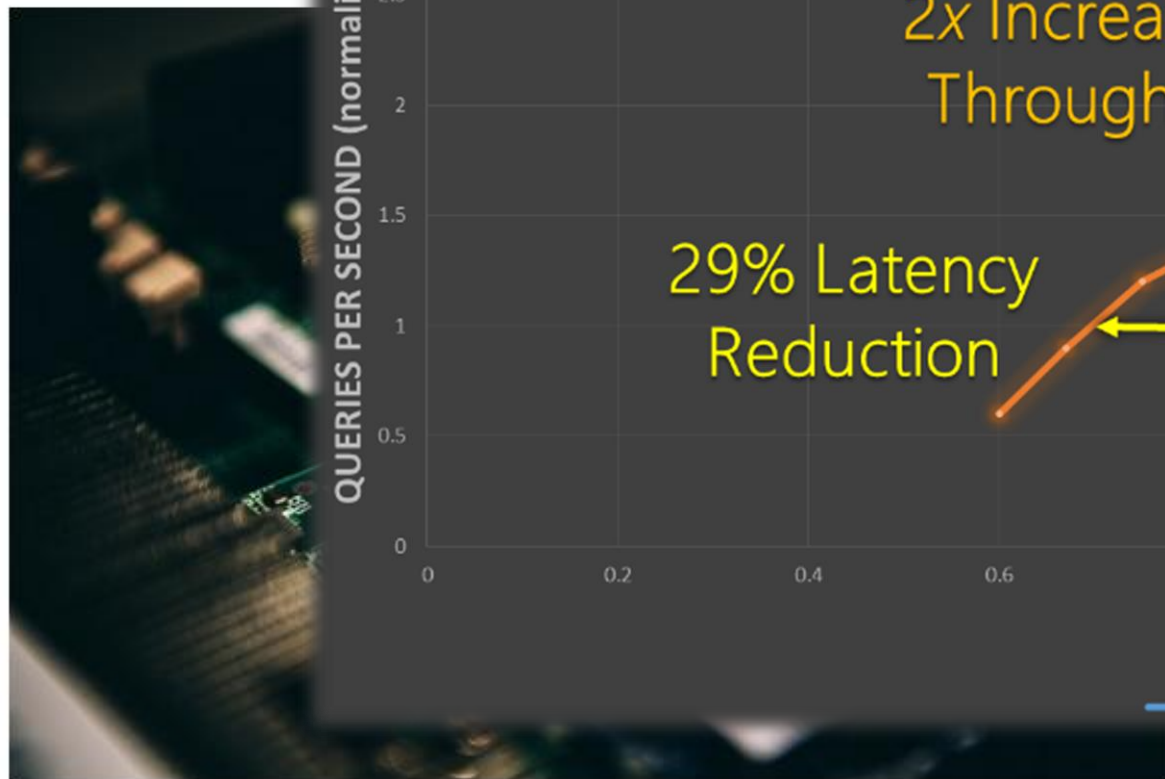
# MICROSOFT SUPERCHARGES BING SEARCH WITH PROGRAMMABLE CHIPS



<http://www.wired.com/2014/06/microsoft-fpga/>

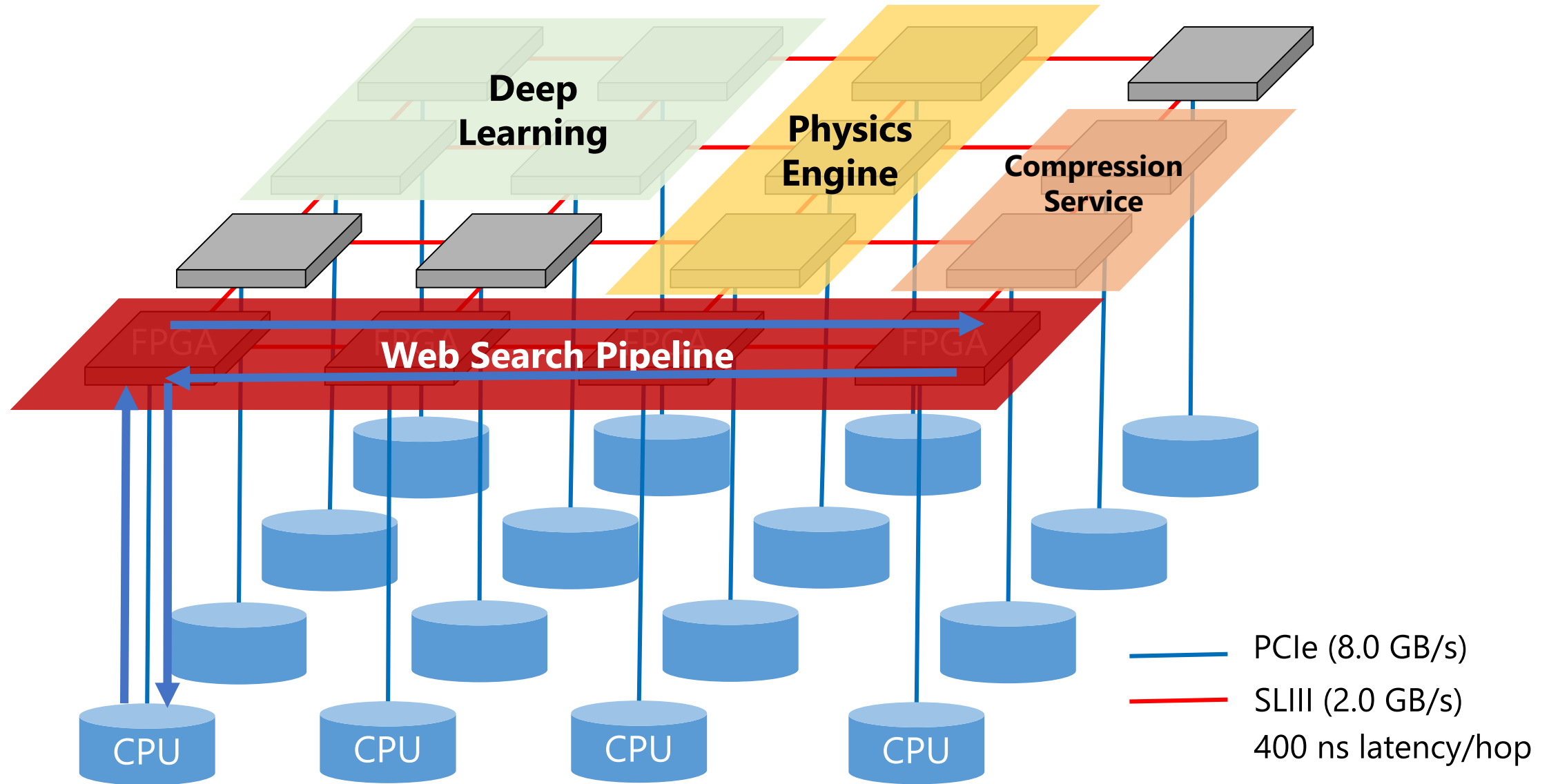


# MICROSOFT SUPERCHARGES BING SEARCH WITH PROGRAM



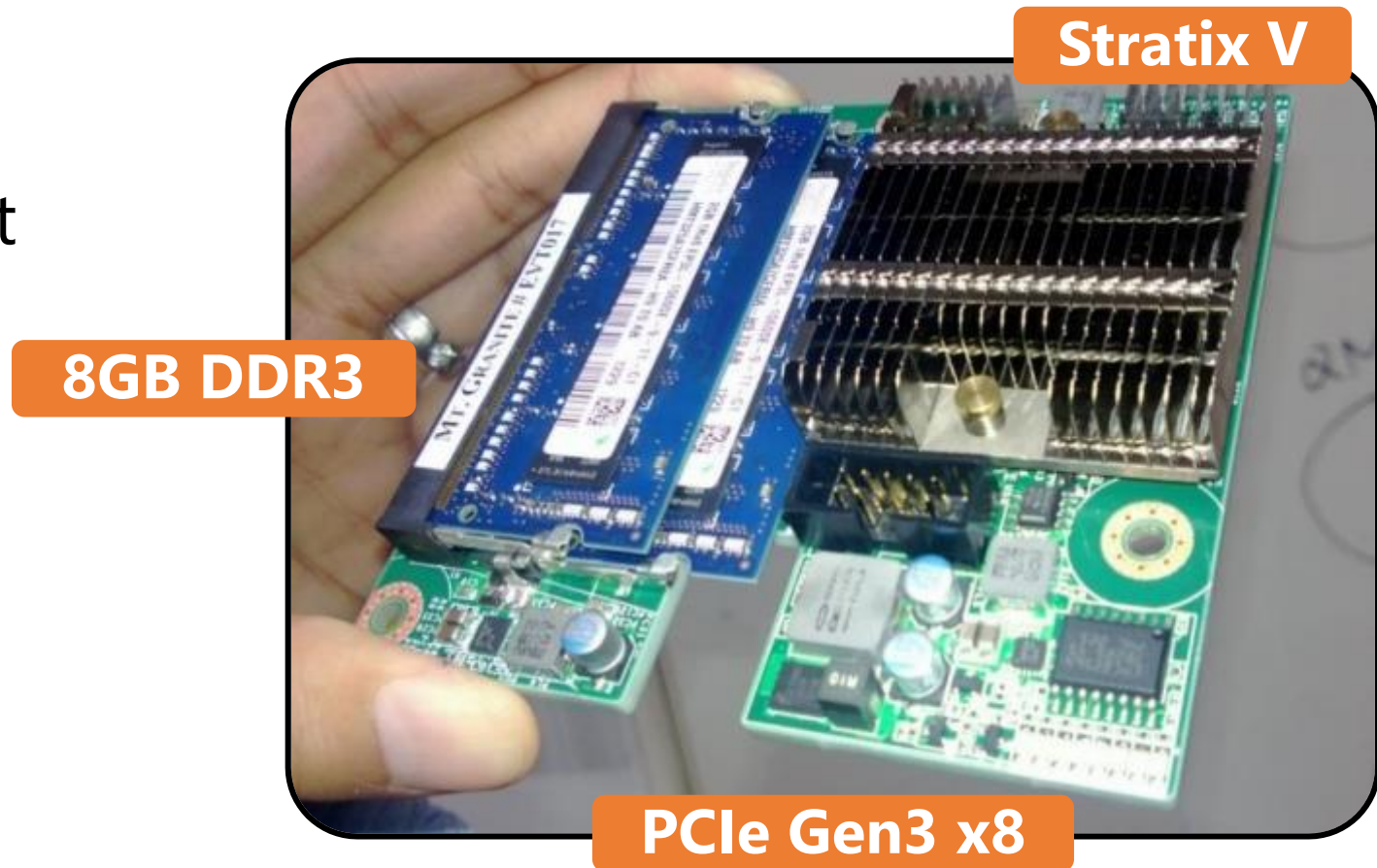
<http://www.wired.com/2014/06/microsoft-fpga/>

# Catapult: An Elastic Reconfigurable Fabric for Datacenters



# Catapult FPGA Accelerator Card

- Altera Stratix V D5
- 172,600 ALMs, 2,014 M20Ks, 1,590 DSPs
- PCIe Gen 3 x8
- 8GB DDR3-1333
- Powered by PCIe slot
- Torus Network



# Microsoft Open Compute Server



- Two 8-core Xeon 2.1 GHz CPUs
- 64 GB DRAM
- 4 HDDs @ 2 TB, 2 SSDs @ 512 GB
- 10 Gb Ethernet
- No cable attachments to server

Air flow

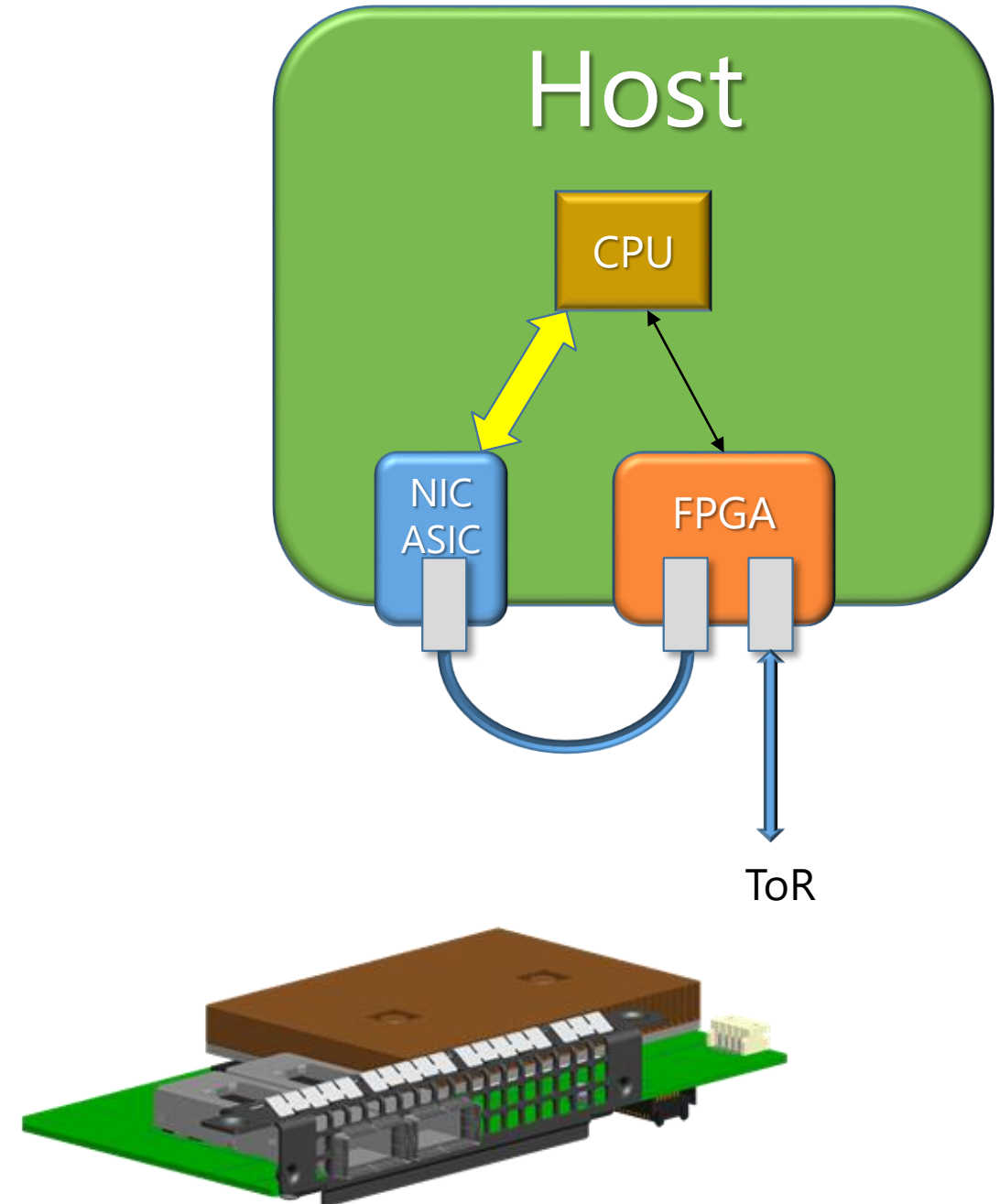
200 LFM

68 °C Inlet



# Azure SmartNIC

- Use Catapult FPGAs for reconfigurable functions
  - Already used in Bing
  - Roll out Hardware as we do software
- Programmed using Generic Flow Tables (GFT)
  - Language for programming SDN to hardware
  - Uses connections and structured actions as primitives
- SmartNIC also does Crypto, QoS, storage acceleration, and more...

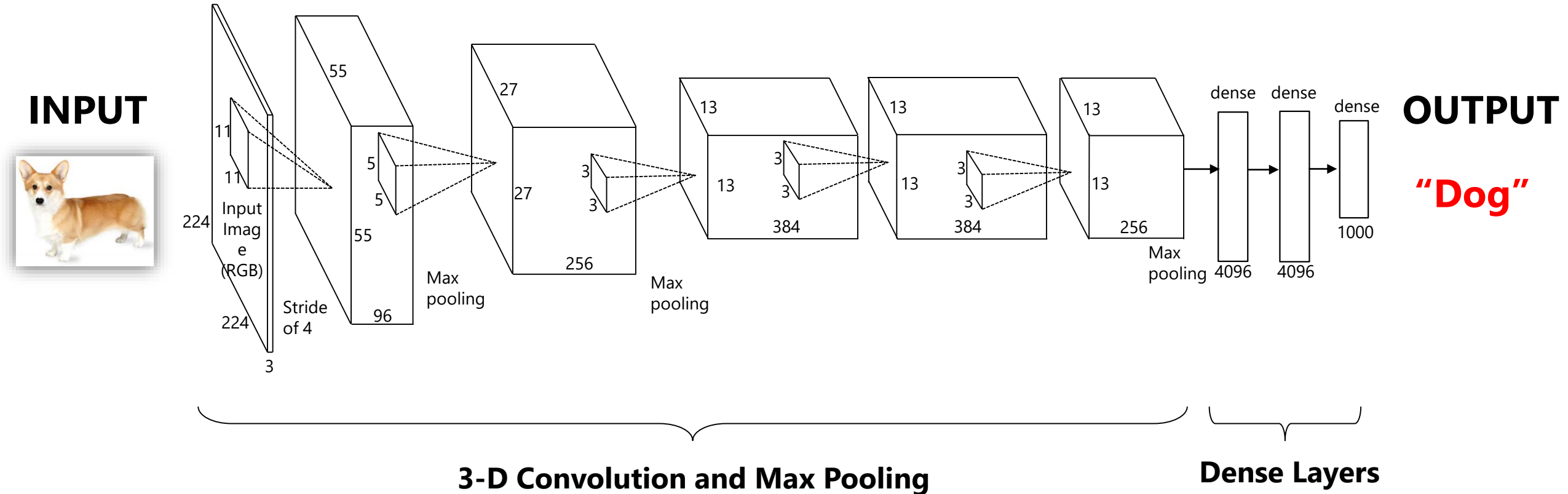


# Harnessing Catapult for Deep CNNs

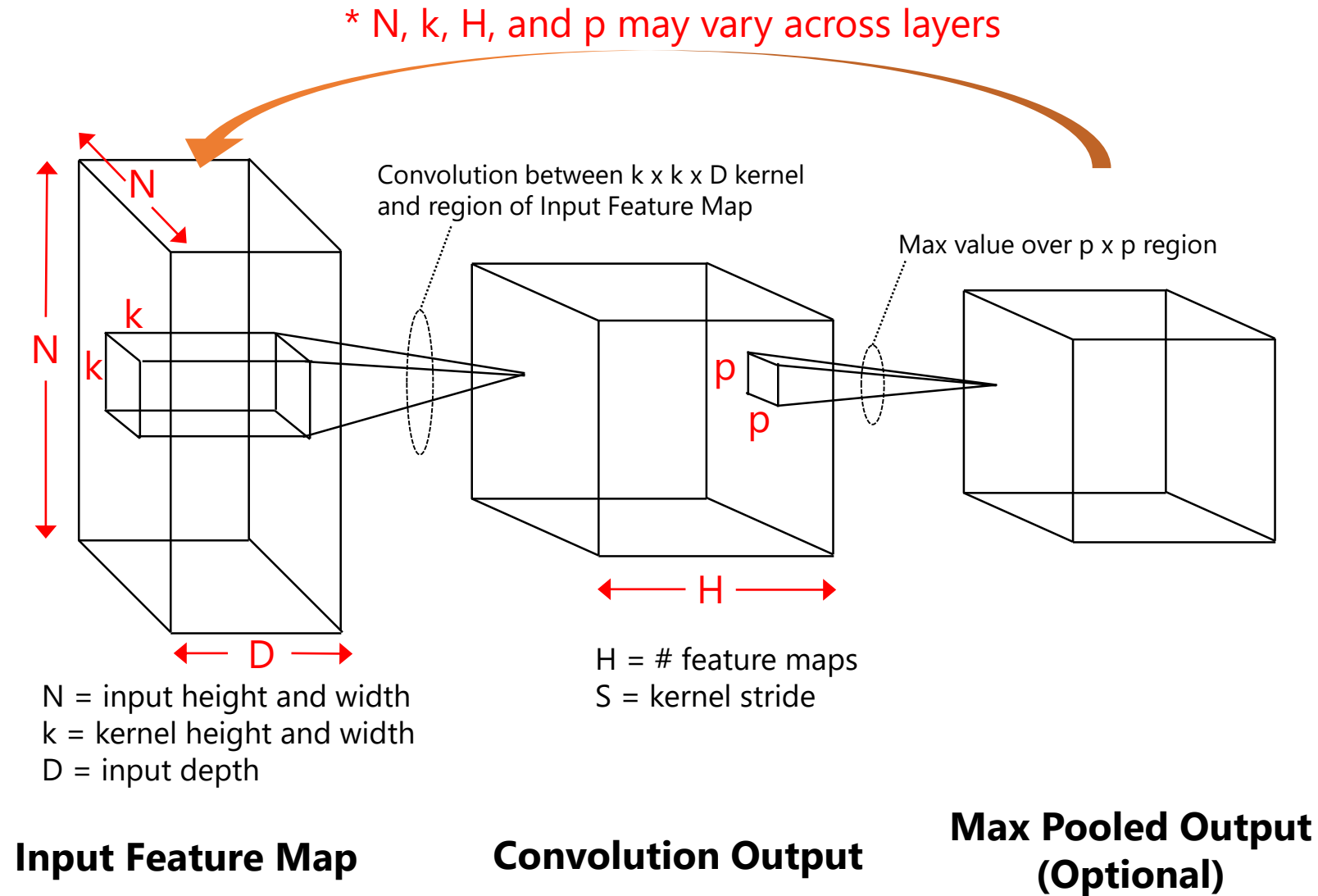
- Leverage abundant FPGA resources in the datacenter for scaling up evaluation and training<sup>1</sup> of deep CNNs
- Achieve order-of-magnitude performance gain relative to CPUs with low cost (<30%) and power (<10%) overheads
- Expose to practitioners as composable SW libraries

<sup>1</sup>*Under development*

# Deep Convolutional Neural Networks

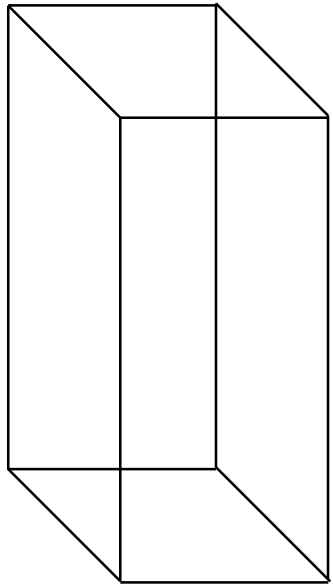


# 3-D Convolution and Max Pooling

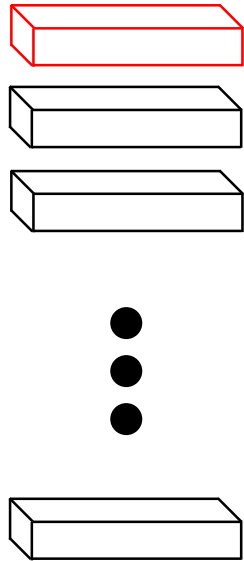




# 3-D Convolution



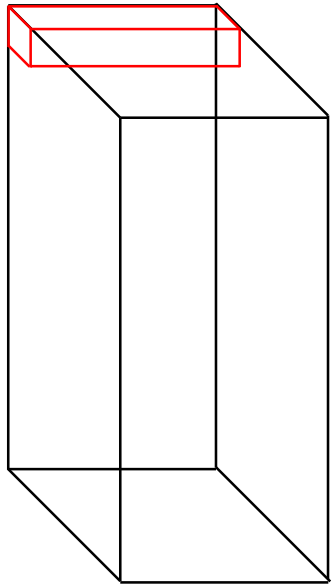
**Input**



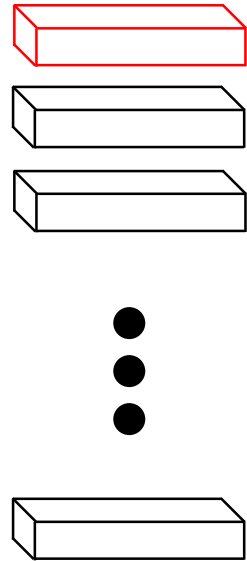
**Kernel  
Weights**

**Output**

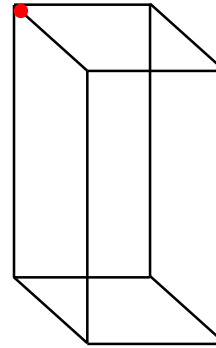
# 3-D Convolution



**Input**

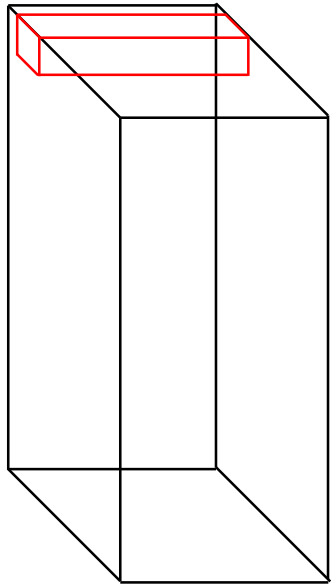


**Kernel  
Weights**

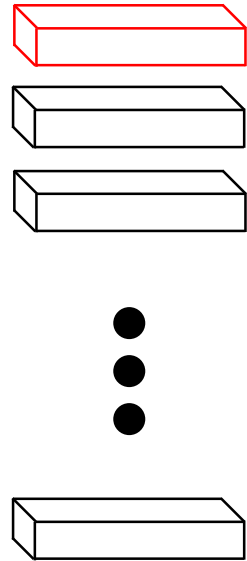


**Output**

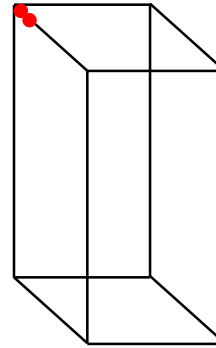
# 3-D Convolution



**Input**

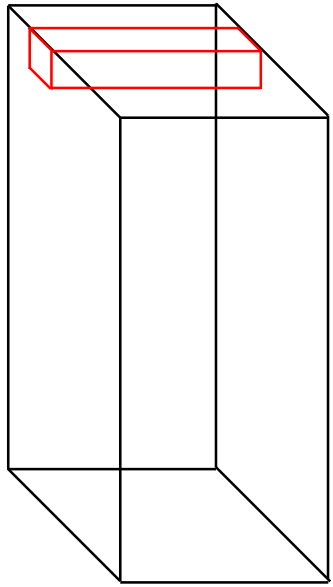


**Kernel  
Weights**

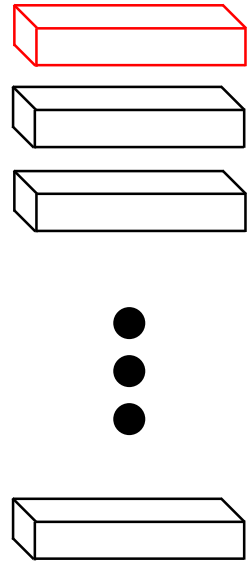


**Output**

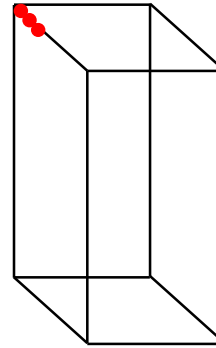
# 3-D Convolution



**Input**

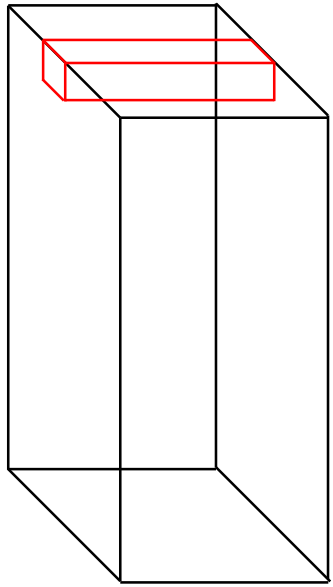


**Kernel  
Weights**

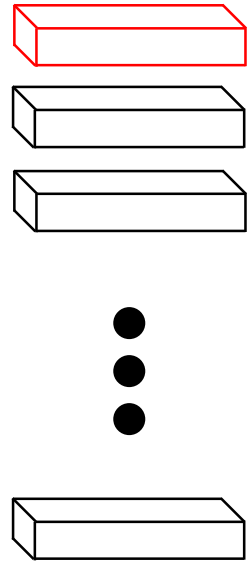


**Output**

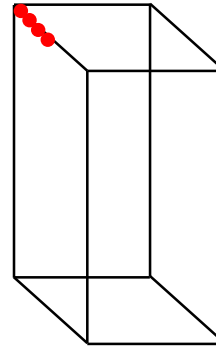
# 3-D Convolution



**Input**

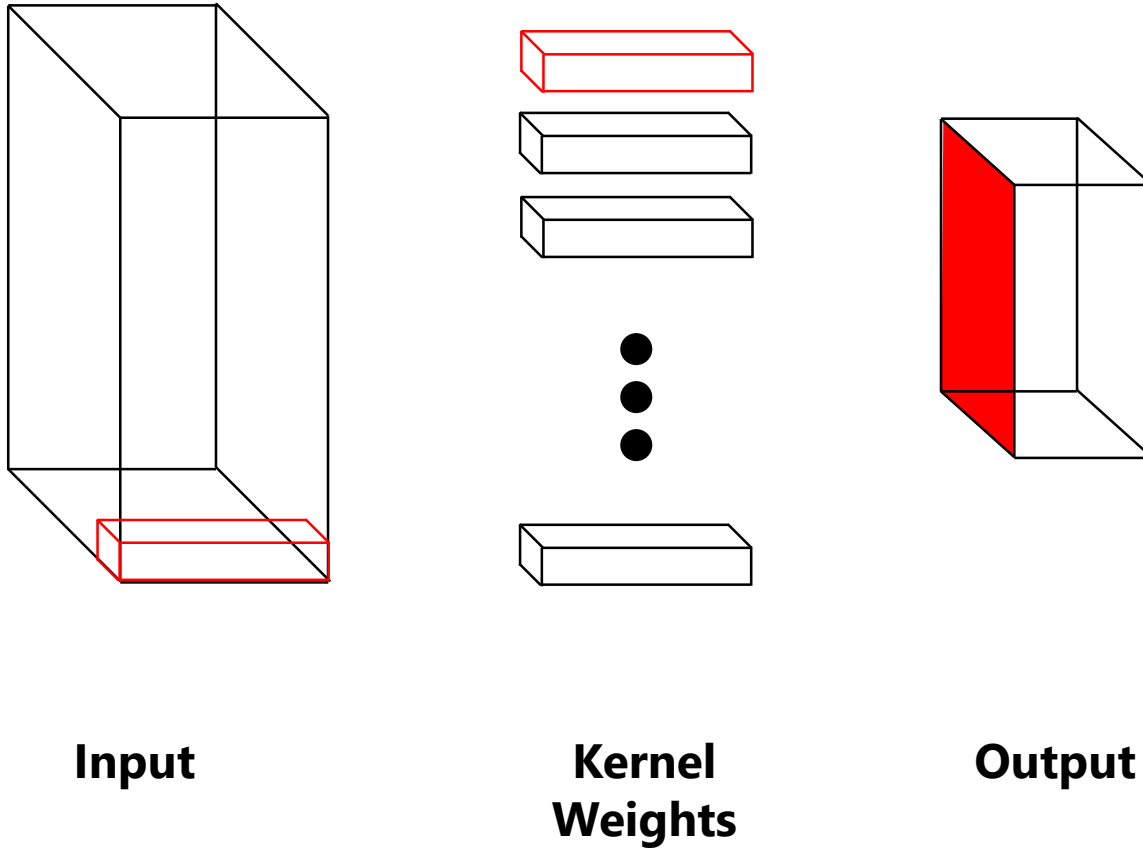


**Kernel  
Weights**

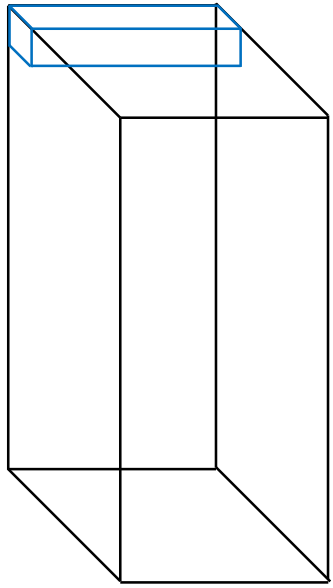


**Output**

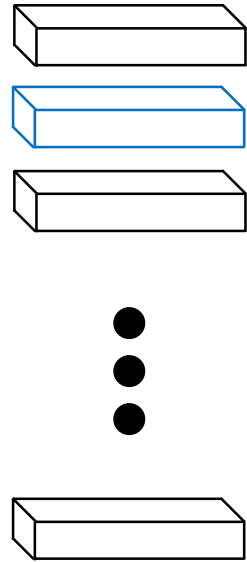
# 3-D Convolution



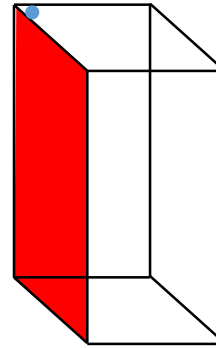
# 3-D Convolution



**Input**

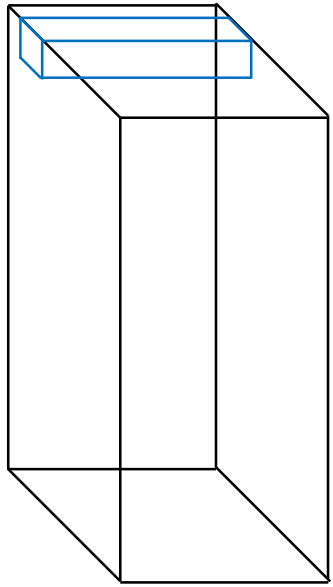


**Kernel  
Weights**

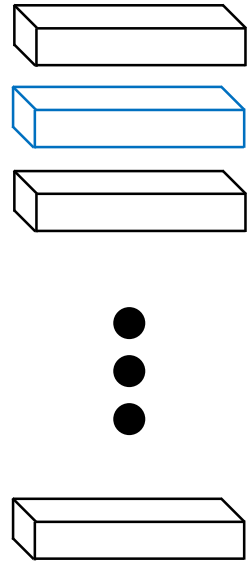


**Output**

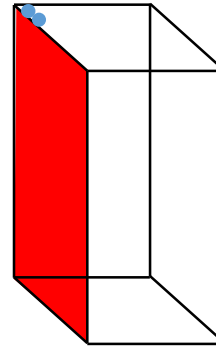
# 3-D Convolution



**Input**



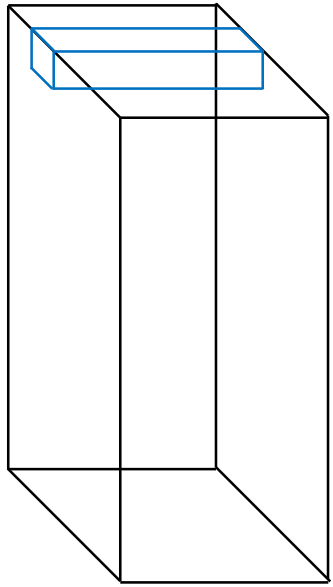
**Kernel  
Weights**



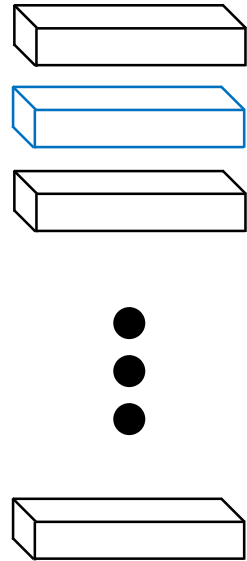
**Output**



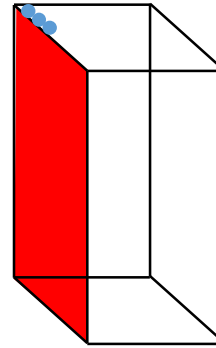
# 3-D Convolution



**Input**

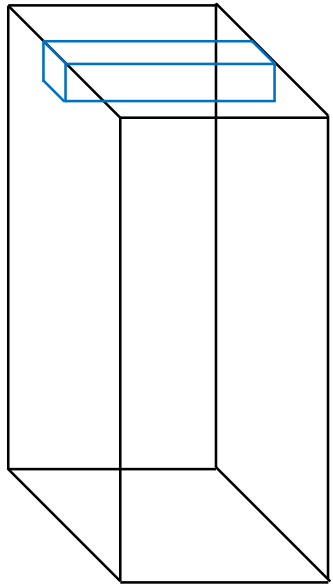


**Kernel  
Weights**

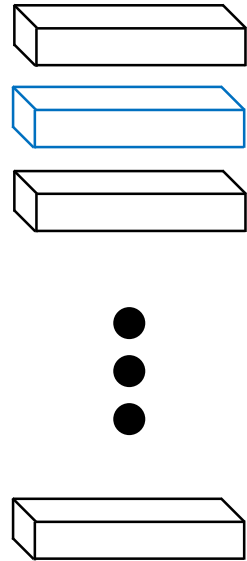


**Output**

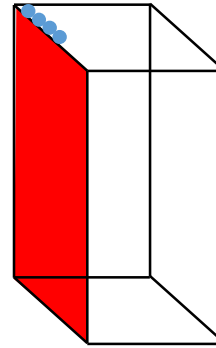
# 3-D Convolution



**Input**

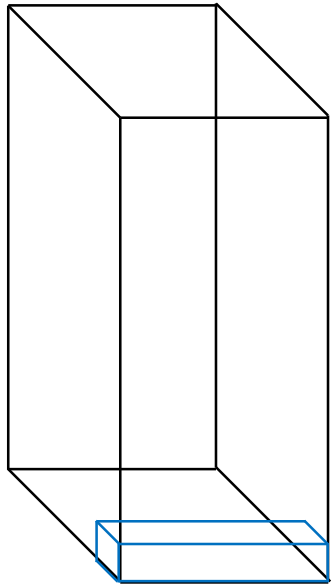


**Kernel  
Weights**

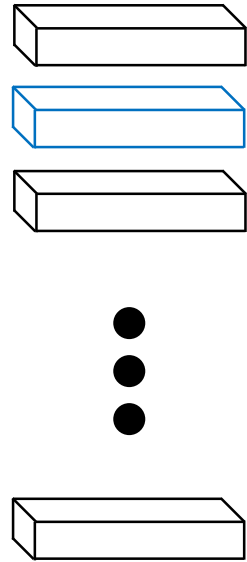


**Output**

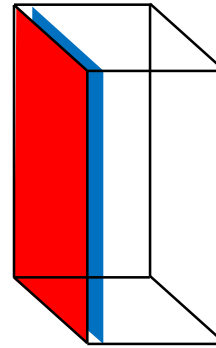
# 3-D Convolution



**Input**

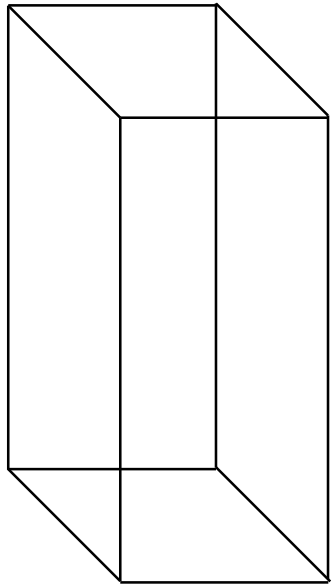


**Kernel  
Weights**

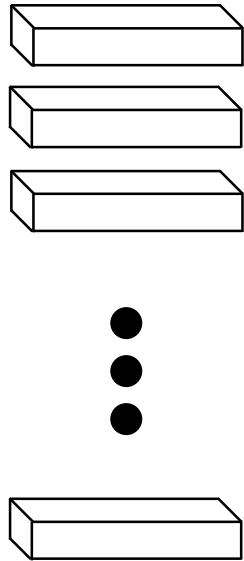


**Output**

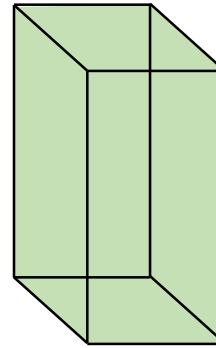
# 3-D Convolution



**Input**



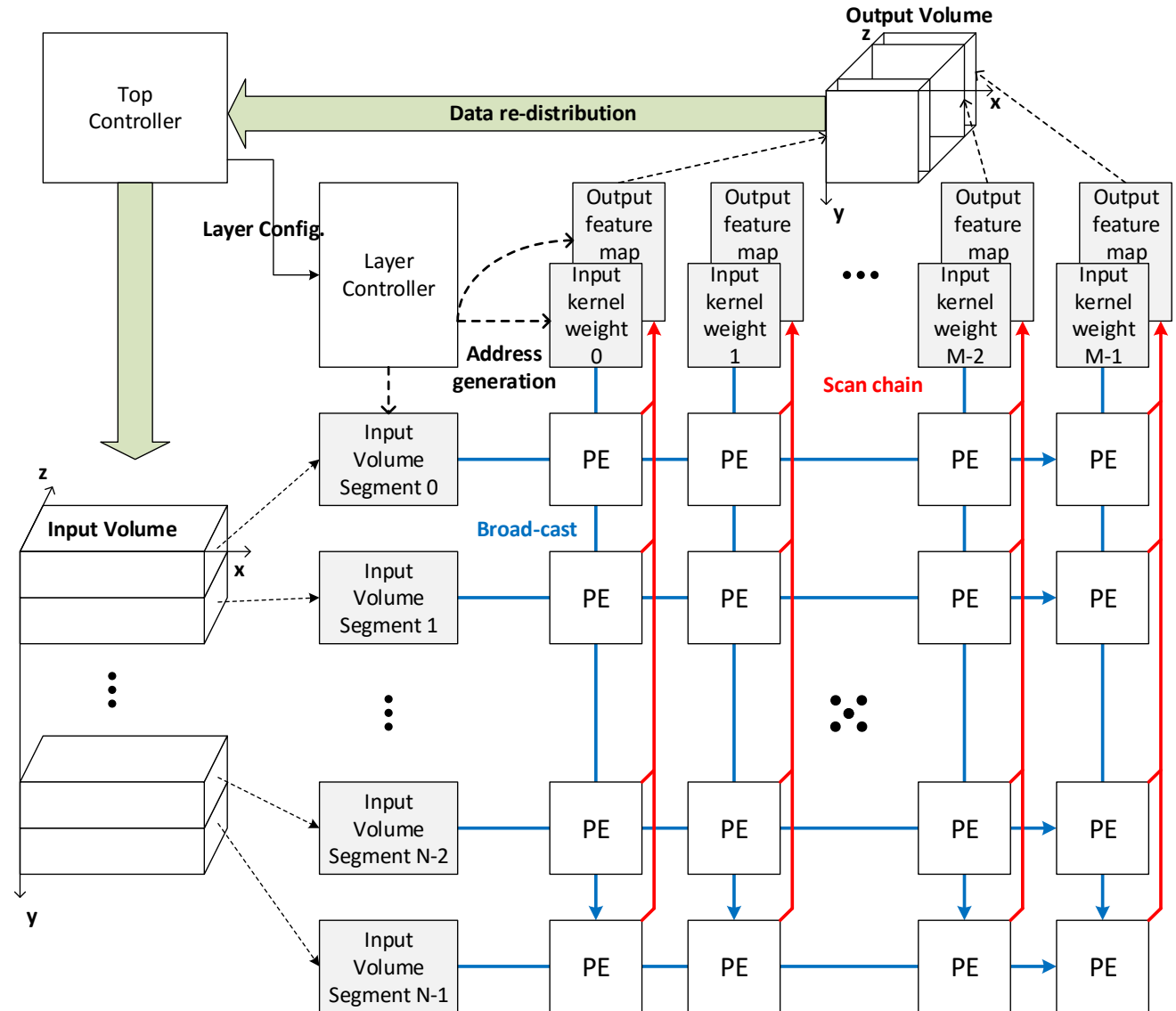
**Kernel  
Weights**



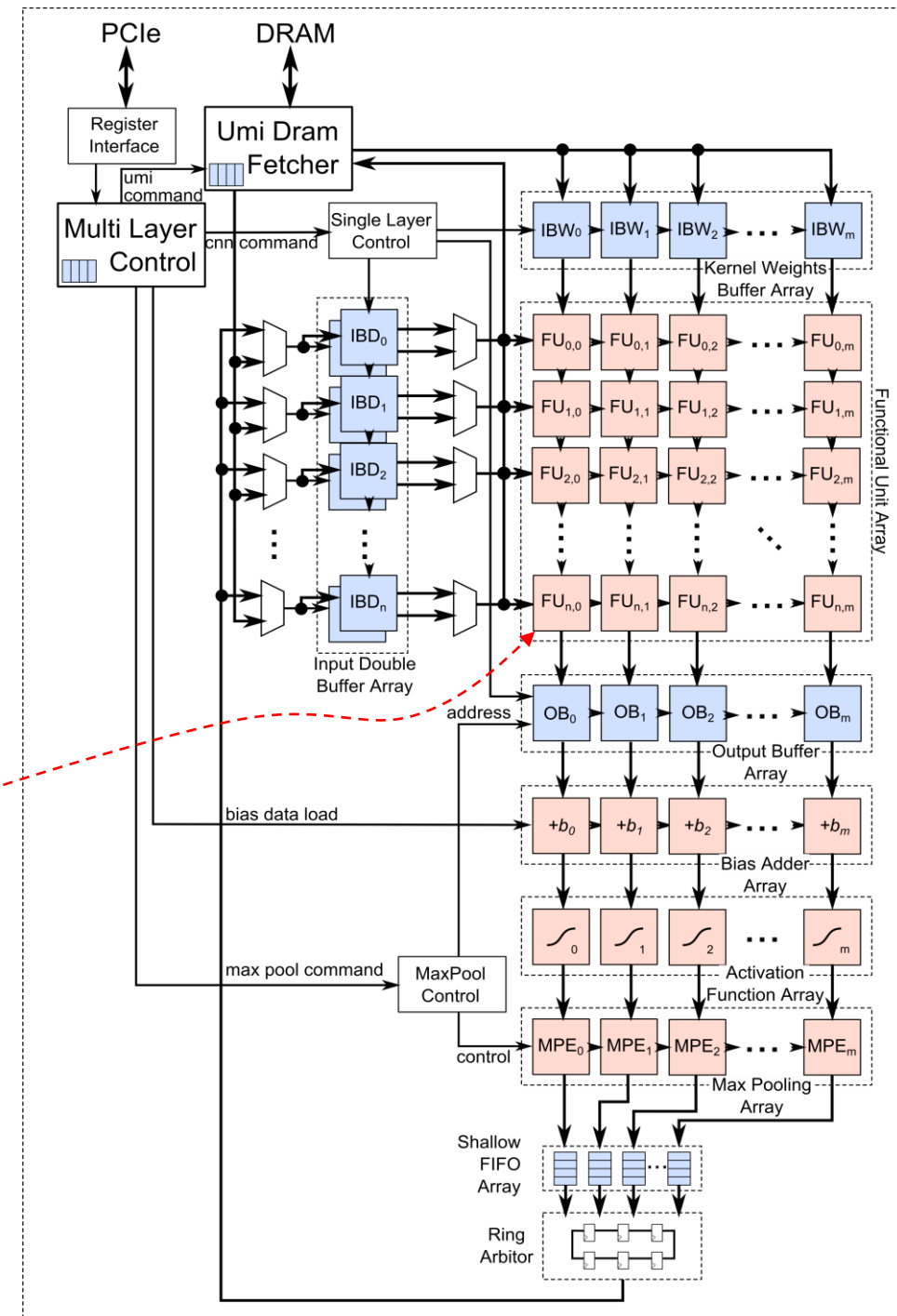
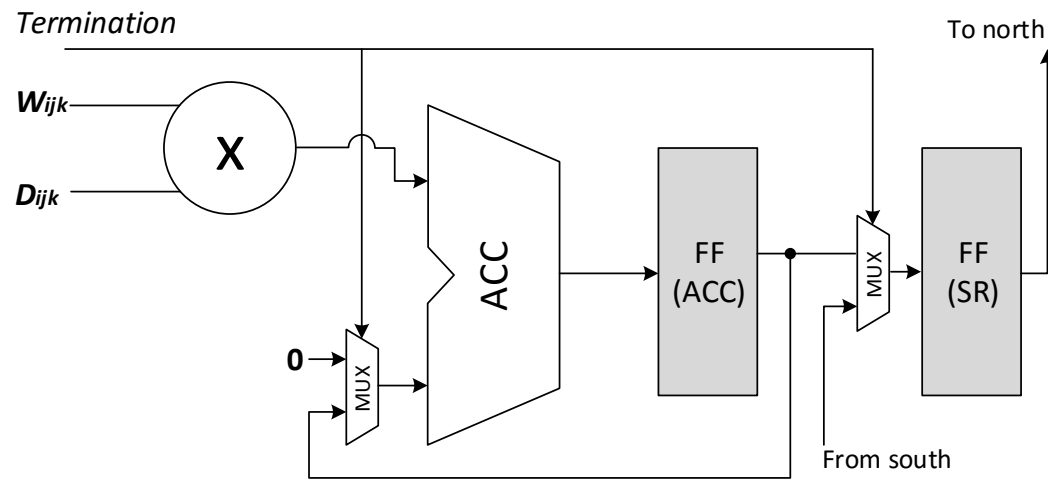
**Output**

# CNN Accelerator Building Block

- Configurable
  - Numerical precision (static)
  - Number of layers
  - Layer dimensions
  - Stride and pooling
- Scalable
  - Can compose multiple engines together over Catapult network
- Efficient
  - Minimize memory bandwidth via data re-distribution NoC
  - On-chip per-row broadcast



# Systolic Array Microarchitecture



DEMO

# ImageNet-1K Classification Performance

| Platform                                      | Library/OS                           | ImageNet 1K Inference Throughput | Peak TFLOPs | Effective TFLOPs | Estimated Peak Power with Server | Estimated GOPs/J (assuming peak power) |
|---|--------------------------------------|----------------------------------|-------------|------------------|----------------------------------|--|
| <b>16-core, 2-socket Xeon E5-2450, 2.1GHz</b> | Caffe + Intel MKL<br>Ubuntu 14.04.1* | 53 images/s                      | 0.27T       | 0.074T (27%)     | ~225W                            | ~0.3                                   |
| <b>Arria 10 GX1150</b>                        | Windows Server 2012                  | 369 images/s <sup>1</sup>        | 1.366T      | 0.51T (38%)      | ~250W                            | ~2.0                                   |

<sup>1</sup>Dense layer time estimated

<sup>2</sup><https://github.com/soumith/convnet-benchmarks>



# ImageNet-1K Classification Performance

| Platform                                      | Library/OS                           | ImageNet 1K Inference Throughput | Peak TFLOPs | Effective TFLOPs | Estimated Peak Power with Server | Estimated GOPs/J (assuming peak power) |
|---|--------------------------------------|----------------------------------|-------------|------------------|----------------------------------|--|
| <b>16-core, 2-socket Xeon E5-2450, 2.1GHz</b> | Caffe + Intel MKL<br>Ubuntu 14.04.1* | 53 images/s                      | 0.27T       | 0.074T (27%)     | ~225W                            | ~0.3                                   |
| <b>Arria 10 GX1150</b>                        | Windows Server 2012                  | 369 images/s <sup>1</sup>        | 1.366T      | 0.51T (38%)      | ~250W                            | ~2.0                                   |
| <b>NervanaSys-32 on NVIDIA Titan X</b>        | NervanaSys-32 on<br>Ubuntu 14.0.4    | 4129 images/s <sup>2</sup>       | 6.6T        | 5.75T (87%)      | ~475W                            | ~12.1                                  |

*Includes server power; however, CPUs available to other jobs in the datacenter*

<sup>1</sup>Dense layer time estimated

<sup>2</sup><https://github.com/soumith/convnet-benchmarks>

# ImageNet-1K Classification Performance

| Platform                                      | Library/OS                           | ImageNet 1K Inference Throughput | Peak TFLOPs | Effective TFLOPs | Estimated Peak Power for <b>CNN Computation</b> | Estimated GOPs/J (assuming peak power) |
|---|--------------------------------------|----------------------------------|-------------|------------------|---|--|
| <b>16-core, 2-socket Xeon E5-2450, 2.1GHz</b> | Caffe + Intel MKL<br>Ubuntu 14.04.1* | 53 images/s                      | 0.27T       | 0.074T (27%)     | ~225W   | ~0.3                                   |
| <b>Arria 10 GX1150</b>                        | Windows Server 2012                  | 369 images/s <sup>1</sup>        | 1.366T      | 0.51T (38%)      | <b>~25W</b>                                     | <b>~20.6</b>                           |
| <b>NervanaSys-32 on NVIDIA Titan X</b>        | NervanaSys-32 on<br>Ubuntu 14.0.4    | 4129 images/s <sup>2</sup>       | 6.6T        | 5.75T (87%)      | <b>~250W</b>                                    | <b>~23.0</b>                           |

*Under-utilized FPGA vs.  
highly tuned GPU-friendly workload*

<sup>1</sup>Dense layer time estimated

<sup>2</sup><https://github.com/soumith/convnet-benchmarks>

# Projected Improvements with Tuning

| Platform                                      | Library/OS                           | ImageNet 1K Inference Throughput                            | Peak TFLOPs | Effective TFLOPs                             | Estimated Peak Power for <b>CNN Computation</b> | Estimated GOPs/J (assuming peak power) |
|---|--------------------------------------|---|-------------|--|---|--|
| <b>16-core, 2-socket Xeon E5-2450, 2.1GHz</b> | Caffe + Intel MKL<br>Ubuntu 14.04.1* | 53 images/s   | 0.27T       | 0.074T (27%)                                 | ~225W   | ~0.3                                   |
| <b>Arria 10 GX1150</b>                        | Windows Server 2012                  | <del>369 images/s<sup>1</sup></del><br><b>~880 images/s</b> | 1.366T      | <del>0.51T (38%)</del><br><b>~1.2T (89%)</b> | ~25W  | <del>20.6</del><br><b>~49.0</b>        |
| <b>NervanaSys-32 on NVIDIA Titan X</b>        | NervanaSys-32 on<br>Ubuntu 14.0.4    | 4129 images/s <sup>2</sup>                                  | 6.6T        | 5.75T (87%)                                  | ~250W   | ~23.0                                  |

*Projected Results Assuming  
Floorplanning and Scaling Up PEs*

<sup>1</sup>Dense layer time estimated

<sup>2</sup><https://github.com/soumith/convnet-benchmarks>

# Are FPGAs a Promising Target in the Datacenter for Deep Learning? **Yes.**

- Best-case FPGA design is  $\sim 1/5$ th GPU throughput but can overtake at scale
- Although CNNs are ideal on GPUs, FPGA achieves GPU-like energy efficiency  $\rightarrow \sim 2X$  energy efficiency within reach
- FPGA is 7X faster ( $\sim 16X$  within reach) than multicore CPUs while flexible enough for diverse cloud scenarios (Bing Ranking, Azure SmartNIC)

# Related Work

- ASICs
  - [Holler'90], [Chen'14], [Cavigelli'15], etc.
- FPGAs
  - [LeCun'09], [Farabet'10], [Aysegul'13], [Baidu'14], [Gokhale'15], [Zhang'15], etc.
- Appliances
  - Nervana, Nvidia DIGITS, Ersatz, etc.

Thank you!  
erchung@microsoft.com